情報処理学会論文誌　コンピューティングシステム　Vol. 2　No. 2　110–121 (July 2009)

光サーキットネットワークの補助的利用による HPC アプリケーション性能向上

滝澤　真一朗[†1]　遠藤　敏夫[†1]　松岡　聡[†1,†2]

多数のノードからなる大規模 HPC システムでは，全ノードを高バンド幅で全対全接続するネットワークは金銭コストや電力消費の問題で実現困難である．我々は低バイセクションバンド幅電気パケット（EPS）ネットワークと高バンド幅光サーキット（OCS）ネットワークからなるネットワーク環境と，その環境での通信手法を提案する．この環境では，各ノードは単一リンクで EPS ネットワークに接続され，一部のノードは OCS ネットワークへも単一リンクで接続される．アプリケーションの通信パターンを考慮して，異なる EPS スイッチに属する OCS ネットワークに接続されたノード間に光回線を割り当て，さらに他ノードからのメッセージを中継させることで，EPS ネットワーク上流で起こりうる混雑を回避する．シミュレーションによる評価の結果，全ノードの半数を OCS ネットワークに接続することで，高いバイセクションバンド幅を要求するアプリケーションでの性能向上，特に全対全通信においては Fat Tree EPS ネットワークと同程度の性能を示すことを確認した．

HPC Application Performance Improvement by a Supplemental Optical Circuit Switching Network

Shin'ichiro Takizawa,[†1] Toshio Endo[†1] and Satoshi Matsuoka[†1,†2]

For large scale HPC systems which consist of many nodes, it will be unfeasible to construct a fully-connected network with high bisection bandwidth due to cost and power consumption, etc. We propose a hybrid network that is composed of an electronic packet switching (EPS) network with low bisection bandwidth and a high bandwidth supplemental optical circuit switching (OCS) network, and communication method on the network. In this network, each node connects to the EPS network with one link and partial nodes also do to the OCS network with another one link. We assign optical pathways to node pairs that are connected to the OCS network and are not in the same EPS switch by considering application's communication pattern. We avoid contentions on the EPS upstream network by letting these nodes relay messages from other nodes. By conducting simulations, we confirmed that our approach can improve the performance of applications which require high bisection bandwidth by connecting only half of nodes to the OCS network. Moreover, performance of all-to-all communication on our system was almost the same as that on fat tree EPS only network.

1. はじめに

マルチコアプロセッサを多数搭載する大規模 HPC システムでは，従来使われていた電気パケット交換方式を採用した高バイセクションバンド幅のクロスバーや Fat Tree などのノード間全対全接続ネットワークは，金銭コストや性能面で実現困難である．そのため，現状では Blue Gene で用いられている 3D トーラスネットワーク[1] のようなノード間接続数の少ないネットワークや，東京工業大学の TSUBAME Grid Cluster（以下 TSUBAME）[2] で用いられているバイセクションバンド幅が低い Tree ネットワークなどが採用されている．TACC Ranger[3] や T2K システム[4] のような高バイセクションバンド幅ネットワークを持つシステムもあるが，将来は並列度のさらなる増加が予想され，そのようなネットワークの規模の維持は困難になると考えられる．

一方で HPC システム上で実行される MPI アプリケーションの多くには通信に局所性があり，各プロセスは一部の特定のプロセスとのみ通信を行うという特徴がある[5]．通信局所性を持つアプリケーションに対しては，高いバイセクションバンド幅を提供しなくても，通信パターンに着目した通信最適化手法を用いることで実行性能向上を実現できる．その手法の 1 つとして，電気パケット（EPS: Electronic Packet Switching）ネットワークと光サーキット（OCS: Optical Circit Switching）ネットワークを組み合わせたネットワーク環境が提案されている[6)–8]．EPS ネットワークとは，HPC システムで広く用いられている Ethernet や InfiniBand からなるネットワークである．OCS ネットワークは以下の特徴を持つネットワーク環境である．

- エンドツーエンドを光信号で通信を行う．
- 回線交換型ネットワーク．通信前後に 2 ノード間で光回線の確立・解放が必要であり，

†1 東京工業大学
Tokyo Institute of Technology
†2 国立情報学研究所
National Institute of Informatics

それには機械処理を要するため，ミリ秒オーダの時間がかかる[9]．

- 広帯域，低遅延，低消費電力．

これらの既存研究ではアプリケーションの通信に合わせて光回線を割り当てることで，EPS ネットワーク上では遠く離れたノード間の通信を OCS ネットワーク上で高速に行うことが可能になる．しかしながら，複数の EPS，OCS ネットワークを必要とするため，ネットワーク規模が大きく構築が容易でない．

大規模 HPC システム用ネットワークとして，我々は各ノードが単一の低バイセクションバンド幅 EPS ネットワークと，単一 OCS ネットワークに接続された EPS-OCS ハイブリッドネットワーク環境を提案し，その環境での MPI 通信手法を提案した[10]．アプリケーション通信パターンに従い EPS スイッチ間通信を行うノード間に光回線を割り当て，大容量の EPS スイッチ間通信を OCS ネットワーク上で中継転送する．これにより，単一の OCS ネットワークであっても，アプリケーションの実行性能がフルバイセクションバンド幅の EPS ネットワークに匹敵することを確認した．この性能は，わずか 1/4 のノードのみ光回線を用いた状況で達成された．残りの 3/4 を使用することでさらなる性能向上を確認したが，微々たるものだった．OCS ネットワークの規模に無駄があり，さらに縮小してもこの性能を維持できると考えられる．

本研究では上記の研究で得た知見を基に，単一の小規模 OCS ネットワークを補助的に使用する EPS-OCS ハイブリッドネットワーク環境を提案する．OCS ネットワークの補助的利用とは，EPS ネットワークに接続されたノードの一部のみを OCS ネットワークに接続することである．さらに，上記研究で提案した，OCS ネットワーク上で中継転送を行う通信手法に対する以下の 3 点の改良を提案する．一部のノードのみ OCS ネットワークに接続されているという制約の下で，アプリケーションの通信パターンを可能な限り満たすように光回線を割り当てる．EPS スイッチ間の通信量に応じて，通信の多い EPS スイッチ間の中継バンド幅を増強すべく，それらスイッチ下のノード間に優先的に光回線を割り当てる．中継ノードの負荷を減らすために，中継ノードの EPS リンクバンド幅を増強する．シミュレーションによる評価の結果，提案システムは EPS ネットワークのバイセクションバンド幅が低い場合でも，全ノードの半数を OCS ネットワークに接続することで，高いバイセクションバンド幅を要求するアプリケーションに効果があることが確認できた．特に全対全通信は Fat Tree EPS ネットワークと同程度の性能であることを確認した．

2. OCS ネットワークの補助的利用の提案

図 1 に OCS ネットワークを補助的に接続したネットワーク環境を示す．EPS ネットワークは既存の HPC システムで用いられいている，InfiniBand などを利用したパケット交換型のノード間相互通信網を表す．図中では 2 階層の Tree トポロジで描かれているが，全ノードが全対全で接続されていればどのようなトポロジでもかまわず，低バイセクションバンド幅であってもかまわない．システム運営に必要となるアカウントなどの情報サービス，ストレージ通信はこの EPS ネットワークを用いて行われるとする．我々は，この EPS ネットワークに属するノードの一部を単一の高バンド幅 OCS ネットワークに接続する．図 1 では，各末端 EPS スイッチ下の 2 ノードが OCS ネットワークに接続されている．OCS ネットワークに接続するノード数に制限を設けた理由は，大多数のノードを OCS ネットワークに接続する場合，任意のノード間で常に回線確立できるようにするには多段に OCS スイッチを組まなければならず，コスト増加につながるからである．以降，OCS ネットワークに接続されたノードを OCS ノードと呼ぶ．

各 OCS ノードは単一リンクで OCS ネットワークに接続されるため，OCS ネットワーク上では 1 度に通信できる宛先ノードは 1 つに限られる．接続に制限のある OCS ネットワークを最大限に活用するため，頻繁に大容量通信を行うノード間に光回線を割り当て，EPS ネットワークのショートカット経路として使用する．OCS ネットワークは任意の OCS ノード間で回線が確立できるよう構成する．また，3 章で述べるように，OCS ノードは同一 EPS スイッチ下の他ノードからのメッセージを他 EPS スイッチ下ノードへ中継転送するため，

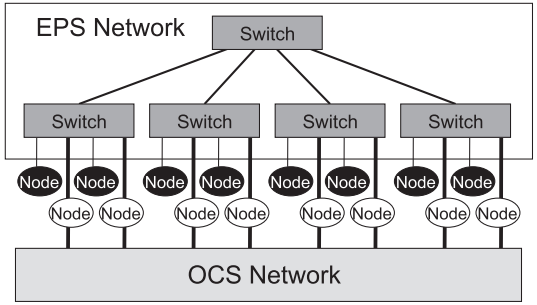


図 1　OCS ネットワークの補助的利用環境
Fig. 1　A network environment that is supplemented by an OCS network.

OCS ノードの EPS リンクバンド幅は他のノードの EPS リンクバンド幅よりも高く構成する．

OCS ネットワークの代わりに別の EPS ネットワークを増設する方法もある．実際，接続に制約のある OCS ネットワークよりも，EPS ネットワークの方がアプリケーションの通信要求に柔軟に対応できるという利点があるが，以下の 2 つの理由より我々は OCS ネットワークを選択した．1 つはスイッチの消費電力である．電気的なパケット処理を高速に行う必要のある EPS スイッチは消費電力が高く，たとえば，Voltaire 社の 288 ポートの InfiniBand スイッチは 2,500 W 消費する．一方，高速な電気処理を行わない OCS スイッチは，GlimmerGlass 社の 190 ポートの製品で 80 W と，上記スイッチの 30 分の 1 以下である．すなわち，OCS ネットワークを用いることで，補助的ネットワークの消費電力を大きくおさえることが可能である．2 つ目の理由は，MPI アプリケーションに通信の局所性があることである．このため，主である EPS ネットワークのスイッチ内に通信の多くを埋め込むことができる．それでも溢れた通信，すなわち EPS スイッチをまたぐ通信が発生するが，それらに対してのみ OCS ネットワークを用いる方法をとれば，接続が制限されていても問題は少ない．以上より，接続が制限されていることの欠点より，消費電力がはるかに少ない利点の方が大きいと判断し，OCS ネットワークを使用することにした．

一方，構築コストの点では，現状 OCS ネットワークの方が高い．OCS スイッチは現在生産・出荷規模が小さいために，小規模スイッチであっても同規模の InfiniBand スイッチの数倍の価格である．しかしながら，EPS スイッチで必要とされる OEO（電気–光–電気）変換器，電気バッファなどの高価な部品を必要としないため，製造コストは高くないと考えられる．将来の価格についての予測はできないが，関連研究を含め，OCS ネットワークを HPC システムで利用する試みは増えているので，近い将来での普及，コモディティ化による価格の下落を期待している．一方で OCS ネットワークに接続するノード NIC としては，LC コネクタなどファイバを使用する 10GbE や 40GbE NIC が使用でき，それらは将来的に InfiniBand などと同等なレベルまで価格が下がることが予想できる．すなわち，使用するスイッチの規模によるが，同一ポート数をサポートするフルバイセクションバンド幅を持つ EPS ネットワークより安価に OCS ネットワークが構築可能になれば，構築コストの問題もなくなる．

3. 提案 MPI 通信手法

OCS ネットワークを補助的に用いた環境での通信の特徴をまとめ，アプリケーション通信パターンを考慮して光回線割当てを行う，通信手法を提案する．

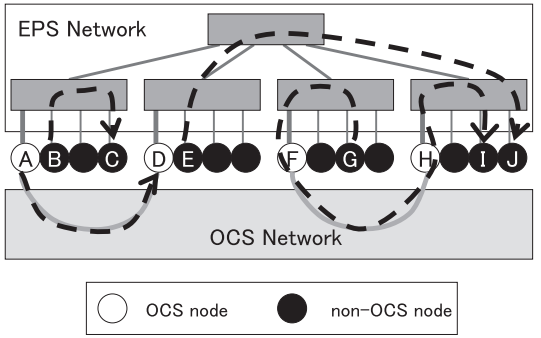


図 2　EPS-OCS ハイブリッドネットワーク上の全通信経路
Fig. 2　All routes over the EPS-OCS hybrid network.

3.1 OCS 補助的利用環境での通信の課題

EPS ネットワークは全ノードにわたり提供されているものの，バイセクションバンド幅が低い場合にはネットワーク上流で混雑が起こりうる．OCS ネットワークでは高バンド幅光回線を用いて任意の OCS ノード間の通信をショートカットできるが，1 度に通信できる宛先ノードは 1 つであり接続に制限があること，回線確立/解放に長時間要することより，複数の通信相手との頻繁な通信には不向きである．さらに，OCS ノードは全ノードの一部であるため，OCS ノード上で大容量通信を行うプロセスが実行されていない限り光回線は有効利用できず，さもなければ，EPS 上流ネットワークに大容量メッセージを送出することになり，帯域圧迫，性能低下へとつながる．EPS，OCS のそれぞれに利点，欠点があるため，互いの欠点を補うよう通信を行う必要がある．

3.2 通信パターンに応じたメッセージ経路

アプリケーションで通信されるメッセージのサイズ，宛先プロセスの位置に応じて異なる通信経路をとる．使用される全経路パターンを**図 2** に示す．なお，この図ではすでに OCS node である A と D の間，および F と H の間に光回線が割り当てられているとする．

メッセージサイズが OCS ネットワークの帯域遅延積よりも小さい場合，メッセージは EPS ネットワークのみを用いて通信される（図中 $B \Rightarrow C$，$E \Rightarrow J$）．帯域遅延積とはリンクを理論的に満たすデータ量であり，送信側がこの値よりも小さいメッセージを送信すると，受信側からの ACK 待ちのための無転送時間が生じ，スループットが低下し，光回線の

バンド幅を十分に活かしきれないためである．また，宛先プロセスが同一 EPS スイッチ下のノード上で動作している場合には，メッセージサイズによらず EPS ネットワークのみを用いた通信を行う（図中 $B \Rightarrow C$）．これには 2 つ理由がある．1 つは単一 EPS スイッチ内の通信であれば，高いスイッチ内バンド幅を活かした高速通信が可能だからである．もう 1 つは，接続に制限のある OCS ネットワークを，すでに十分ノード間距離の近い EPS スイッチ内の通信に用いるには贅沢だからである．

メッセージサイズが OCS ネットワークの帯域遅延積より大きく，宛先プロセスが他 EPS スイッチ下ノード上で動作している場合にのみ，OCS ネットワークを用いる．大容量通信を行う OCS ノード間を光回線で接続し，その通信を OCS ネットワーク上で行う（図中 $A \Rightarrow D$）．さらに，OCS ノードでないノード間の大容量通信を OCS ノードが中継する（図中 $G \Rightarrow F \Rightarrow H \Rightarrow I$）．このように，従来なら EPS ネットワーク上流に流れていたメッセージを OCS ネットワーク側に追いやることで，EPS ネットワーク上流の帯域圧迫を避け，混雑による遅延を回避する．さらに EPS スイッチ間に複数の光回線を割り当てることで，スイッチ間通信のバンド幅増強も行える．一方で OCS ノードの EPS リンクが混雑しうるが，バンド幅を増強することで緩和する．末端リンクであるため，EPS 上流リンクのバンド幅を増強することより容易であると考えている．

3.3　大容量通信のための経路作成方法

以上の通信手法を実現するためには，大容量通信用の経路を作成する必要がある．経路作成には，アプリケーションの通信パターン，プロセスの配置情報を基に，OCS ノード間で光回線を確立し，ノード間のルーティングテーブルの計算が必要となる．以降 3.3.1 項から 3.3.3 項にかけて経路作成方法を説明する．

この経路作成の実行手段は 2 種類ある．1 つはアプリケーションをあらかじめプレ実行して必要な情報を取得し，本実行前に経路を確定する方法である．もう 1 つはアプリケーション実行中に，最初に 1 度，あるいは定期的に作成する方法である．この方法は特にパラメータを変えて繰り返し処理を行うアプリケーションに対して有効であり，通信パターンに変更がない限り，第 1 イテレーションの間に取得した情報を基に経路を作成し，第 2 イテレーション以降終了までその経路を使い続けることが可能となる．

3.3.1　プロセス配置情報，通信パターンの取得

プロセス配置情報として，各プロセスを実行しているノードの ID（EPS ネットワーク上の IP アドレスなど）を取得し，ノード，およびプロセスを EPS スイッチ単位でグルーピングする．また，OCS ノードの ID も取得する．前者はノード間通信を行う際に EPS スイッチ内で閉じた通信か，EPS スイッチ間通信かを判断するために，後者はどのノードが光回線を用いた通信を行えるのか判断するためである．

通信パターンとして，プロセスごとに OCS ネットワークの帯域遅延積以上のサイズのメッセージ送信の受信相手と，その相手への平均送信メッセージサイズを取得する．さらに 1 ノード上で複数プロセスを実行している場合，プロセス間の通信パターンからプロセス配置情報を基に，ノード間の通信パターンを抽出する．

```
入力
Onodes    : OCS ノードの集合
CommPaths : EPS スイッチをまたぐ OCS ノード間の
            通信の集合
出力
OnPairs   : 光回線で接続する OCS ノードペアの集合

1: while (Onodes または CommPaths が空でない)
2:   SwPair = 光回線で接続する EPS スイッチのペア
              を選択
3:   OnPair = 光回線を割り当てる OCS ノードのペア
              を選択 (Onodes, CommPaths,
                      SwPair)
4:   OnPairs   << OnPair
5:   Onodes    -= OnPair 内 2 ノード
6:   CommPaths -= OnPair が関係する通信
7: end
```

図 3　回線割当てアルゴリズム
Fig. 3　Path assignment algorithm.

3.3.2　光回線の割り当て

3.3.1 項で取得したプロセス配置と通信パターンを基に，可能な限り通信パターンを満たすように OCS ノード間で光回線を確立する．光回線割当てアルゴリズムの概要を Ruby-like な文法で記述したものを**図 3** に示す．

Onodes は MPI プロセスを実行する OCS ノードの集合である．CommPaths はこれら OCS ノード間で行われる通信のうち，EPS スイッチをまたぐ通信の集合である．各要素は，通信を行う 2 ノードの ID，通信量からなる．

OCS ノード間に光回線を割り当てる際には，まず OCS ネットワーク側で接続する EPS スイッチのペアを定める（2 行目）．続いて，実際に光回線を割り当てる OCS ノードのペア

を，先ほど求めた EPS スイッチのペア，使用可能な OCS ノードの集合，考慮すべき OCS ノード間通信パターンを基に決定する（3 行目）．2 行目，3 行目の処理の詳細は後述する．選ばれた OCS ノードペアを記録（4 行目）後，光回線で接続された OCS ノードにはこれ以上光回線を割り当てることができないので，OCS ノードの集合から除外する（5 行目）．同様に，OCS ノードが複数の EPS スイッチ間通信を行うとしても，接続された相手以外とは直に OCS ネットワーク上で通信を行えないため，すべての関係する EPS スイッチ間通信を集合から除外する（6 行目）．

以上の処理を終了条件となるまで繰り返す（1 行目）．Onodes が空となる状況は，すべての OCS ノード間で光回線が割り当てられた場合である．CommPaths が空となる状況は，アプリケーションの通信を満たすように十分な数の光回線が割り当てられた場合，あるいは，宛先 OCS ノードがすでに他の OCS ノードと接続済みで回線を確立できない場合である．後者の状況となった場合には光回線が割り当てられない OCS ノードが生じるが，これ以上通信パターンを満たす必要がない，また満たせないため，使用しない．

以下に 2 行目，3 行目の詳細を述べる．

EPS スイッチペアの選択　あらかじめノード間通信パターンから各 EPS スイッチ間の通信数を求めておく．これは，2 つの EPS スイッチ間の通信を行うノードペアの数である．まず，通信を行う EPS スイッチ間に 1 本ずつ光回線が割り当てられるように，EPS スイッチ間通信数の少ない順にペアを選択する．これにより，後続する OCS ノードペア選択時に，実際に通信を行う OCS ノードペアが選択される可能性が向上する．通信を行うすべての EPS スイッチ間に最低 1 本の光回線が割り当てられた後は，「EPS スイッチ間通信数 – 割当て済み光回線数」を計算し，通信の多い EPS スイッチ間バンド幅を増強すべく，この値の大きい EPS スイッチペアを選択する．

OCS ノードペアの選択　選択された EPS スイッチ間通信を行う OCS ノードペアを，ノード間通信量（メッセージサイズ）の多い順に選択する．過去の OCS ノードペア選択状況によって，選択された EPS スイッチ間通信を行うノードペアを選択できない場合もある．その場合は，該当する 2 つの EPS スイッチ下の OCS ノードから，光回線が割り当てられていないものを 1 つずつランダムに選び出し，そのペアを選択する．

すなわち，通信量の多い OCS ノードペア間に優先的に光回線を割り当てることで，ボトルネック通信の最適化を行っている．また，通信量の多い EPS スイッチ間に数多くの回線を割り当てることで，中継転送の効率を最大化している．

```
入力
Topo    : EPS ネットワークトポロジ情報
Olinks  : 割当て済み光回線の集合
出力
ルーティングテーブル

1: 初期ルーティングテーブルを作成 (Topo, Olinks)
2: (EPS スイッチ数).times do
3:   if (OCS ノードである)
4:     光回線で接続されているノードと経路を交換
5:     ルーティングテーブルを更新
6:   end
7:   同一 EPS スイッチ下の OCS ノードから経路情報を
     取得
8:   ルーティングテーブルを更新
9: end
```

図 4　経路作成アルゴリズム
Fig. 4　Route creation algorithm.

3.3.3　通信経路の作成

以上で割り当てた光回線を用いた，EPS-OCS 両ネットワークにわたる大容量メッセージの通信経路を作成する．各ノードのルーティングテーブルを**図 4** に示すアルゴリズムより計算する．

まず初期ルーティングテーブルを作成し，1 ホップで到達できるノードへの経路を記録する（1 行目）．すなわち，同一 EPS スイッチ下のノードへの経路と，OCS ノードの場合，光回線で接続された相手ノードへの経路である．以降，OCS ノードから経路情報を取得し，ルーティングテーブルの更新を繰り返し，各宛先ノードへの最小ホップ経路を求める（2〜9 行目）．ここで EPS スイッチ数分繰り返すのは，複数の光回線を使用したホップ数の長い通信経路を考慮するためである．しかしながら，そのような経路は遅延が大きくなるために除外し，繰返し回数を削減することも可能である．光回線に接続された OCS ノードの場合，接続されたノードと経路情報を交換し，それを基に経路更新する（3〜6 行目）．各ノードは，同一 EPS スイッチ下の OCS ノードの経路情報を取得し，経路を更新する（7〜8 行目）．

繰返し回数を減らした場合や，OCS ネットワークの規模が小さく十分な光回線を割り当てることができない場合には，ノード間に中継経路を作成できないことがある．経路が存在しない場合は大容量通信であっても，図 2 中の経路 $E \Rightarrow J$ のような，EPS ネットワーク

上流を用いた通信を行う．

4. 評　　価

OCS ネットワークを補助的利用した環境と提案通信手法の組合せと，EPS ネットワークのみを用いた場合，フルバイセクションバンド幅かつノンブロッキングである Fat Tree トポロジの EPS のみのネットワークとの実行時間比較をシミュレーションによって行う．本提案の評価を行う際には，通信パターンを基に光回線を割り当て，フォワーディングテーブルを作成済みとした．

4.1 実 験 設 定

4.1.1 アプリケーション

2 次元格子上の隣接通信と，NAS Parallel Benchmarks（NPB）の CG，IS，LU[11] の 4 アプリケーションを用いた．256 プロセスを使用し，NPB の 3 アプリケーションでは問題サイズを C とした．2 次元格子上の隣接通信では，16×16 の格子に配置した各プロセスが隣接する最大 4 プロセスとそれぞれ 4 MB のメッセージを交換しあう．格子の第 1 行第 1 列から行方向に順にランク（MPI プロセス ID）を割り振った．各アプリケーションは同じ計算の繰返し処理からなっており，本実験では 5 イテレーションまで実行した．IS は全対全通信を繰り返し実行する，局所性のないアプリケーションである．

4.1.2 シミュレーション環境パラメータ

64 ノードからなる環境を想定してシミュレーションを行った．EPS ネットワークのトポロジとしては図 1 と同様な 2 階層の Tree トポロジを用いた．末端 EPS スイッチ構成は，16 ポートスイッチ 4 機とした．OCS ネットワークの規模として，OCS ノード数 4，8，16，32，64 の 5 通りを用いた．EPS スイッチあたりの OCS ノード数をバランスし，各スイッチ以下 1，2，4，8，16 と配置した．

詳細なシミュレーションパラメータを**表 1** に示す．CPU core speed は後述するシミュレータで用いるためのアプリケーション MPI 関数トレースを取得した環境の CPU 速度である．各ノード 4CPU core からなるとし，1 ノード上で 4 プロセスを実行した．EPS ネットワークの上流リンクのバンド幅として，提案ネットワークと EPS ネットワークのみの場合（表中 Ours & EPS only）は 20 Gbps とした．上流リンクはストレージなど他の通信でも用いられているとし，バンド幅を小さくすることで仮想的な混雑状況を表現した．OCS ノードの EPS リンクのバンド幅は，OCS リンクと同じ 20 Gbps とした．OCS ネットワークの帯域遅延積は 10,000 bit となる．

表 1　シミュレーション環境パラメータ
Table 1　Simulation system parameters.

Parameter	Value
Node Parameter	
CPU core speed	2.0 GHz
Number of cores	4
Propagation delay of intra-node comm.	100 ns
Bandwidth of intra-node comm.	68 Gbps
EPS Network Parameter	
One link propagation delay	500 ns
Bandwidth of upstream link (Ours & EPS only)	20 Gbps
Bandwidth of upstream link (Fat Tree EPS only)	160 Gbps
Bandwidth of downstream link	10 Gbps
Bandwidth of OCS node's downstream link	20 Gbps
MTU	4,096 Bytes
OCS Network Parameter	
Propagation delay	500 ns
Bandwidth	20 Gbps
MTU	4,096 Bytes

4.1.3 プロセス配置

次の 2 パターンの配置を用いた．

Sequential　各 EPS スイッチ下，各ノード上にプロセスを連続配置する．すなわち，プロセス 0 から 63 を 1 つの EPS スイッチ下に配置し，プロセス 0 から 3 を 1 ノードに，プロセス 4 から 7 を別の 1 ノード上で実行する．実行する MPI プロセスのランクの合計が小さいノードを OCS ノードとする．

CommPattern　アプリケーションの通信パターンからプロセス間通信グラフを作成し，4 分割し，グループごとに EPS スイッチ下に配置する．さらに，グループごとにプロセスを 16 分割し，1 ノードに 4 プロセス配置する．グラフ分割ライブラリ metis[12] を用い，グループ間の通信量が最小となるように分割した．EPS スイッチをまたぐ通信量が多いノードを OCS ノードとする．

評価環境上でそれぞれの配置を行った場合の，各アプリケーションの EPS スイッチ間通信量を**表 2** にまとめる．large は OCS ネットワークの帯域遅延積よりも大きいサイズのメッセージの総和を，small は小さいものの総和を，total はその 2 つの合計を表す．LU 以外の

表 2　各アプリケーションの EPS スイッチ間通信量
Table 2　Inter-EPS-switch communication valumes of each application.

Name	Size	Sequential	CommPattern
隣接通信	total	1,920 MB	1,480 MB
	large	1,920 MB	1,480 MB
	small	0 B	0 B
CG	total	2,142 MB	2,544 MB
	large	2,142 MB	2,544 MB
	small	50 KB	212 KB
IS	total	3,462 MB	3,462 MB
	large	3,461 MB	3,461 MB
	small	1.2 MB	1.2 MB
LU	total	151 MB	112 MB
	large	108 MB	79 MB
	small	43 MB	33 MB

アプリケーションでは，通信の 99%が大きいメッセージであると分かる．提案環境を用いた場合，これらメッセージは OCS ネットワーク上を流れる．CG では CommPattern 配置の方が通信量が増えている．metis はヒューリスティックに基づいてグラフ分割を行うために，最適に分割できるとは限らず，精度に問題があるためである．

4.1.4　シミュレータ

評価のためにシミュレータを作成した．このシミュレータは，MPI アプリケーションを実機で実行したときに取得した MPI 関数トレースを入力として，環境パラメータをあてはめて再生し，実行時間を求める．MPI 関数トレースとして，各関数へ渡された引数と，呼び出し時刻，終了時刻を記録する．

シミュレータは CPU 処理，MPI 通信処理をトレースファイル中のすべての関数トレースを処理しきるまで繰り返す．CPU 処理時間として，MPI 関数呼び出し時刻から直前の MPI 関数の終了時刻を差し引いた値を用いた．MPI 通信処理時間は次のように計算した．

1 対 1 通信の場合，リンクごとの「遅延 + メッセージサイズ/バンド幅」の和を通信時間とした．OCS ネットワーク上での通信は，光回線を確立すればスイッチ内では待ちは発生しないため，単一リンクとして処理した．EPS ネットワークでは，各 EPS スイッチは Store-and-Forward 方式とし，EPS スイッチ内通信であれば 2 リンク，EPS スイッチをまたぐ通信であれば 4 リンクとして処理した．また，ノードおよび EPS スイッチでメッセージを受信する際に，同一宛先にメッセージが集中したときの混雑をシミュレートするため，先に受信したメッセージの時刻を考慮し，後続するメッセージの受送信を遅らせた．ただし，

表 3　図 2 における各経路の通信コスト
Table 3　Cost formulas of each route in Fig. 2.

Route	Cost
$B \Rightarrow C$	$2(\alpha_1 + n/\beta_1) + 2f(x,t)$
$A \Rightarrow D$	$\alpha_3 + n/\beta_3 + f(x,t)$
$E \Rightarrow J$	$2(\alpha_1 + n/\beta_1) + 2(\alpha_2 + n/\beta_2)$
	$+4f(x,t)$
$G \Rightarrow F$	$4(\alpha_1 + n/\beta_1) + \alpha_3 + n/\beta_3$
$\Rightarrow H \Rightarrow I$	$+5f(x,t)$

Fat Tree EPS ネットワークの場合は，スイッチ間通信においては混雑が発生しないとし，ノードでのメッセージ受信時の混雑のみを考慮した[⋆1]．MTU サイズを超えるメッセージは MTU サイズに収まるように分割し，複数回送信処理を行う．EPS，OCS ネットワーク間の中継を行う際には，分割されたメッセージをすべて受信した後に中継する．以上より，メッセージサイズを n，EPS 下流ネットワークの遅延，バンド幅をそれぞれ α_1，β_1，EPS 上流ネットワークでは α_2，β_2，OCS ネットワークの場合は α_3，β_3 としたとき，図 2 における各経路の通信時間は**表 3** となる．ここで，$f(x,t)$ はメッセージが経由する装置 x（EPS スイッチ，計算ノード）における，ある時刻 t での混雑を表すパラメータであり，メッセージ受信のたびに通信時間に加えられる．

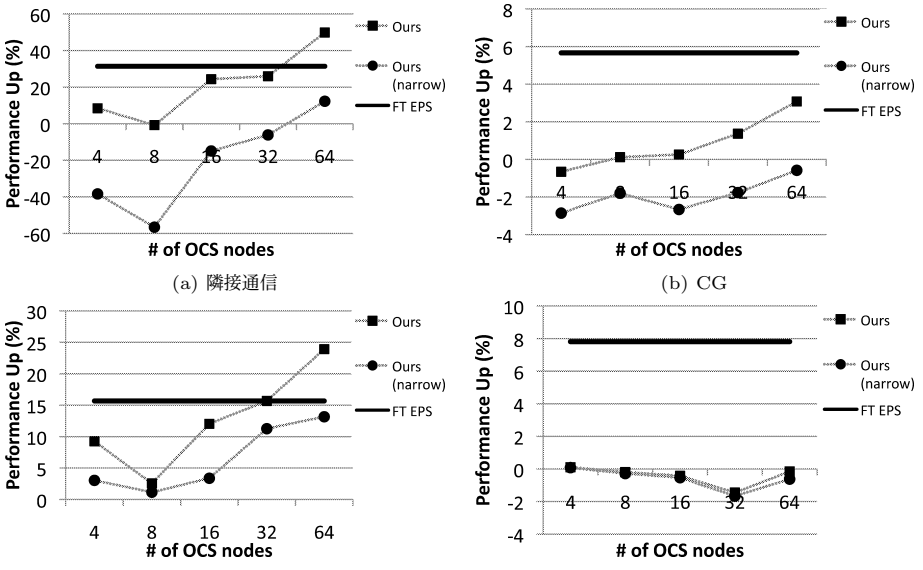
集団通信の場合，MPICH2 で用いられているアルゴリズムどおりに 1 対 1 通信の組合せとして通信時間を計算した [13]．

4.2　結　　果

図 5 に Sequential 配置の，**図 6** に CommPattern 配置での各アプリケーションの結果を示す．各グラフにおいて，横軸は OCS ノード数を，縦軸は EPS ネットワークのみを用いた場合の実行時間に対する，各ネットワークでの実行時間の短縮率を表す．値が大きいほど短い時間で実行を終えたことを表し，処理速度の向上を意味する．凡例「Ours」は提案システムを，「Ours (narrow)」は提案システムにおいて，OCS ノードの EPS リンクバンド幅を他のノードとそろえた場合を，「FT EPS」は Fat Tree EPS ネットワークを表す．

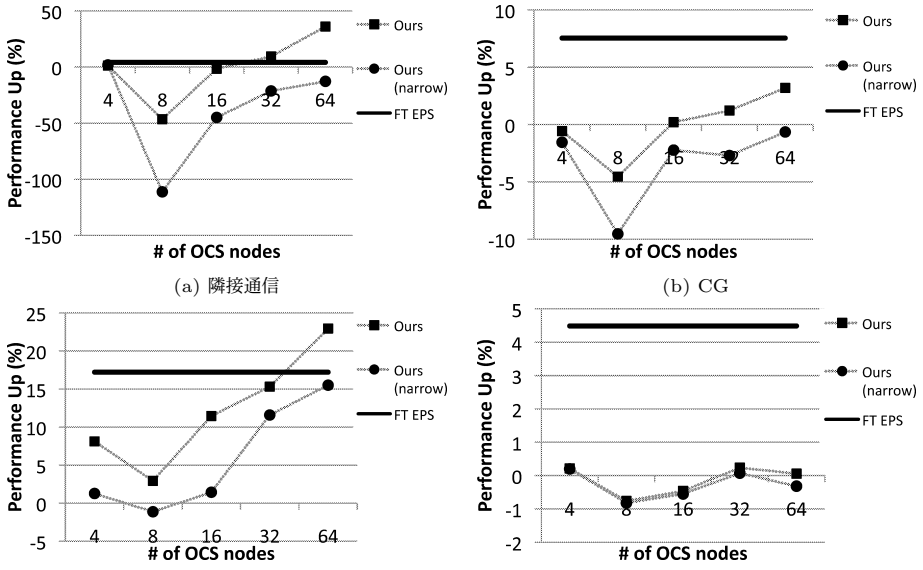
どのアプリケーション，どちらの配置においても，Ours (narrow) は最悪の性能となった．IS 以外の結果では，EPS のみの場合より劣る．これは中継を行う OCS ノードの EPS リンクで混雑が発生しているためであり，そのバンド幅を増やした場合の結果である Ours との

⋆1 すべてのスイッチ間通信経路で混雑が発生しないと仮定したため，性能を高く見積もっている可能性がある．



(a) 隣接通信　(b) CG　(c) IS　(d) LU

図 5　Sequential 配置での EPS のみのネットワークに対する性能向上率
Fig. 5　Performance gain against EPS-only network under the Sequential task allocation.



(a) 隣接通信　(b) CG　(c) IS　(d) LU

図 6　CommPattern 配置での EPS のみのネットワークに対する性能向上率
Fig. 6　Performance gain against EPS-only network under the CommPattern task allocation.

性能差から確認できる．唯一 LU の結果（図 5，6 中 (d)）だけは差が小さいが，LU は表 2 より，EPS スイッチ間通信量が少ないために，中継ノードにかかる負荷も少ないからである．以降では，Ours と FT EPS の比較を行う．

図 5 (a) の隣接通信の結果より，OCS ノード数が全ノード数の半数の 32 の場合，Sequential 配置では EPS に対して 26%の性能向上が確認できた．しかしながら，FT EPS の方が 31%の向上と，効果が大きい．一方，図 6 (a) の CommPattern 配置の場合は，FT EPS は 4%の向上しか得られていないが，Ours では 9%の向上が得られた．この Fat Tree EPS ネットワークの向上率の劇的な減少の理由は，EPS 上流リンクを用いた通信量が減り，上流での混雑が減ったため，EPS のみのネットワークの実行時間が大きく短縮されたことによる．提案システムでは EPS スイッチ間通信を行うノードどうしを光回線で直接接続し，低遅延，高バンド幅通信を行うことで，さらなる性能向上を実現している．このように，通信を考慮してプロセス配置をすることで性能向上することは知られていたが，本提案を用いることでさらなる性能向上が実現できることが確認できた．

隣接通信の結果の特徴的な振舞いとして，OCS ノード数 8 より，回線数の少ない 4 の方が性能が優れている点があげられる．これは以下の理由による．OCS ノード数が 4 の場合，使用できる光回線数は 2 本のため，すべての EPS スイッチ間通信を OCS ネットワーク上で行うことができず，一部の通信は EPS 上流リンクを用いて行われる．一方，OCS ノード数が 8 の場合，4 本の光回線でより多くの EPS スイッチ間を接続でき，より多くの EPS スイッチ間通信を OCS ネットワーク上で行える．特に Sequential 配置の場合にはすべての EPS スイッチ間通信が OCS ネットワークに移譲された．しかしながら，OCS ノードでの中継混雑や，宛先ノードに到着するまでのホップ数の増加により性能低下が起こる．CommPattern 配置で OCS ノード数 8 のとき 50%近い性能低下しているが，このときの最長経路は 3 つの EPS スイッチをまたぐものであり，遅延の大きい経路を選択したためである．グラフ分割の精度の問題で，逆に Sequential 配置よりも通信を行う EPS スイッチペ

アが増えてしまったことによる．

この振舞いは，以降で説明する NPB の各アプリケーションでも確認された．上記の問題は OCS ノード数をさらに増やすことで解決できる．EPS スイッチ間の OCS ネットワーク上でのリンクの本数，バンド幅の増強となるためである．なお，EPS 上流リンクのバンド幅がさらに小さい場合には，このような性能の逆転現象は起こらないと考えられる．

NPB の結果では，EPS のみの場合に対し，IS のみ大幅に性能向上し（図 5，6 (c)），そのほかでは効果が少ない．IS において OCS ノード数 32 のとき，Sequential 配置では FT EPS と同等の 16%の向上であった．CommPattern 配置では 15%の向上であり，FT EPS の 17%には若干劣るものの，同程度の向上であった．IS は全対全通信を行うため，高いバイセクションバンド幅が要求される．OCS ノード数を増やすことで，提案システムでは複数の光回線で EPS スイッチ間を接続することになり，EPS スイッチ間バンド幅の増強，および複数のノードで中継を行うことによる混雑分散により性能を向上させている．しかし，これだけでは 64 ノードすべてを OCS ネットワークに接続しても FT EPS ほどのスイッチ間バンド幅は確保できず，また，中継ノードでの混雑のため，FT EPS を超えることはできない．OCS ノード数 32 以上の Ours が FT EPS 以上の結果となったのは，OCS ノードの EPS リンクバンド幅を増強したことによる．全対全通信のため，各ノードは他の全 63 ノードからメッセージを受信しなければならず，ノード NIC にボトルネックがある．提案システムでは，OCS ノードの中継負荷を減らすために EPS リンクのバンド幅を増強したが，IS の場合では全対全通信の受信ボトルネック解消にも働き，結果として FT EPS を上回ることになった．

CG（図 5，6 (b)）は IS 同様に EPS スイッチ間通信量は多く（表 2），高いバイセクションバンド幅が要求されるアプリケーションだと分かる．実際，IS 同様に OCS ノード数を増やすことで性能が向上している．IS と違い，FT EPS の性能に達しないのは，CG では各プロセスは平均 5 プロセスのみからメッセージを受信するため，ノードの EPS リンクの混雑が少ないためである．

LU（図 5，6 (d)）はそもそも EPS スイッチ間通信量が少ないため（表 2），EPS のみの場合に対する提案システムの効果は少なく，むしろ，中継転送による遅延がバンド幅増強による効果を打ち消し，性能が落ちている場合もある．FT EPS が他より優れている理由は，LU は小さいサイズのメッセージを多く通信するため，EPS のみの場合と Ours では EPS ネットワーク上流で混雑が発生するが，FT EPS は発生しないからである．

以上より，提案システムは高いバイセクションバンド幅を要求するアプリケーションに対して有効であり，特に全対全通信を得意とするといえる．一方でそうでないアプリケーションに対しては，OCS ネットワークの規模を大きくしても効果は少なく，逆に中継遅延による性能低下も起こりうる．また，今回評価は行わなかったが，embarrassingly parallel なアプリケーションに対しては，通信が少ないために OCS ノード数に関係なく，EPS のみの場合に対する性能向上はないと考えられる．これは Fat Tree EPS ネットワークを構築した場合でも同様である．本提案の隣接通信，IS の結果より，全ノードの半数だけを OCS ネットワークに接続することで，Fat Tree EPS ネットワークと同程度の性能を発揮することが確認できた．このとき，CG，LU でも EPS のみの場合に対する性能向上は確認できた．EPS 上流リンクがストレージや他のノード，他のアプリケーションによる通信で圧迫されていて帯域が制限されている場合に有効でるといえる．

5. 構築コスト，消費電力に関する考察

EPS のみのネットワークに対する，本提案を用いて半数のノードを OCS ネットワークに接続した環境，Fat Tree EPS ネットワーク環境の構築コスト，消費電力についての考察をする．なお，考察を簡潔にするために以下の仮定をおく．EPS ネットワークは 2 階層 Tree とする．2 階層で Fat Tree が構築できる十分な数のポートを搭載した EPS スイッチが存在するとする．提案環境において OCS ノードの EPS リンクバンド幅を増強する際は，EPS スイッチ側に十分な数の空きポートがあり，ノード NIC 増設かつトランキングで行うとする．ケーブリングコストは無視する．

n ノードからなる EPS のみのネットワークを k 機の下段スイッチ，l 機の上段スイッチから構成した場合のネットワークコストは以下のように定式化できる．

$$n \times EPS\_NIC + (k+l) \times EPS\_SW \tag{1}$$

EPS_NIC，EPS_SW はそれぞれノード NIC，EPS スイッチのコスト（価格，消費電力）を表す．Fat Tree を構成するには上段にも k 機のスイッチが必要となるため，コストは以下になる．

$$n \times EPS\_NIC + 2k \times EPS\_SW \tag{2}$$

提案システムでは，半数のノードへの EPS NIC の追加，OCS ネットワークの追加が必要で，以下になる．

$$(n + \frac{n}{2}) \times EPS\_NIC + (k+l) \times EPS\_SW + \frac{n}{2} \times OCS\_NIC + OCS\_NET \tag{3}$$

OCS_NIC，OCS_NET はそれぞれノードの OCS ネットワークへの NIC，OCS ネットワークのコストを表す．以上より，式 (2) から (1) を引いた差分である式 (4) が Fat Tree EPS を構築する際に要する追加コスト，式 (3) から (1) を引いた式 (5) が提案システムを構築する際にかかるコストとなる．

$$(k-l) \times EPS\_SW \tag{4}$$

$$\frac{n}{2} \times EPS\_NIC + \frac{n}{2} \times OCS\_NIC + OCS\_NET \tag{5}$$

各種パラメータをあてはめたとき，式 (5) の値が式 (4) の値を下回れば，高いバイセクションバンド幅を要求するアプリケーションに対して，提案システムがコストパフォーマンスに優れた手法だといえる．

構築コストの点では，現状は Fat Tree を構成する場合の方が優れている．その理由は 2 章で述べたとおり，OCS ネットワークを構築する部品，特にスイッチが高価だからである．

一方，消費電力の点では現状でも提案システムの方が優れている．具体的に，上記モデルを TSUBAME にあてはめて考える．TSUBAME では上段に 2 機，下段に 6 機の，既述の 288 ポート InfiniBand スイッチを使用している．Fat Tree 構成にするには上段にさらに 4 機のスイッチが必要となり，このスイッチの消費電力は 2,500 W なので，全体で $4 \times 2{,}500$ で 10 KW（10,000 W）要する．提案システムの場合には，TSUBAME は 655 ノードからなるので半数の 328 ノードに EPS，OCS NIC を搭載する．また，OCS ネットワークは既述の 190 ポート OCS スイッチ 4 機から構成できる．そのスイッチの消費電力は 80 W，最近の 10 Gbps，20 Gbps の InfiniBand HCA や 10GbE NIC の消費電力が 10 W 前後であることから，追加 NIC の消費電力を一律 11 W とすると，$2 \times (328 \times 11) + 4 \times 80$ で 7,536 W となり，Fat Tree より小さくなる．

この消費電力における利点は，今後 HPC システムがさらに大規模化した場合に顕著になる．多数のノードを EPS ネットワークで接続するために，管理やケーブリングの都合上，ポート数の多いスイッチが用いられるが，スイッチ内でフルバイセクションを実現するために多数のチップを必要とするため，小規模スイッチに対しポート数の増加割合以上に消費電力の増加の割合が大きくなる．一方で OCS スイッチ内では回線割当て時にスイッチ内で経路を変更する場合以外はほとんど電気処理を行わないため，大規模スイッチでも低い消費電力が期待できる．さらに，すでに 1000 ポート以上の OCS スイッチを作成する技術も存在し[9]，使用スイッチ数の削減につながる．ノード数が増えるため NIC 数も増えるが，その増加割合に対する消費電力の増加割合は等しく，Fat Tree の場合を超えることはない．そのため，多数の EPS スイッチを要する Fat Tree EPS ネットワークに対し，提案システムははるかに少ない消費電力で構築できる．

6. 関 連 研 究

EPS ネットワークと OCS ネットワークを利用するネットワークは他にも数多く提案されている[6)–8)]．Barker らは，各ノードが低バンド幅 EPS ネットワークと，複数の OCS ネットワークに接続するネットワークを提案している[6)]．EPS ネットワークは小規模メッセージ通信，集団通信に使用し，OCS ネットワークは 1 対 1 の大規模メッセージ通信に使用される．また，同ネットワーク上で複数 OCS ネットワークにわたり各ノードが大規模 1 対 1 メッセージをフォワードする手法も提案している[7)]．Kamil らは低バンド幅 EPS ネットワークとノードとの間に OCS ネットワークを挿入した構成を持つハイブリッドネットワーク HFAST を提案している[8)]．従来ならプロセスマイグレーションが用いられていたところを，光回線を割り当て直すことで，通信を行う任意の 2 ノードを同一 EPS スイッチ下に配置し，通信最適化を行う．HFAST は 1 対 1 の大規模メッセージ通信に使用され，小規模メッセージ通信，集団通信には別の EPS ネットワークが使用される．これら既存研究では複数の EPS，OCS ネットワークを使用する．特に全ノードを OCS ネットワークに接続する必要がある．必然的にネットワーク規模が大きくなり，金銭コスト，消費電力が大きくなる問題がある．また，複数のネットワークにメッセージが分散されるため，個々のネットワークの利用効率が我々の提案よりも低くなる．

過去の研究で我々は，上記既存研究のネットワーク規模の問題を解決するべく，単一 EPS ネットワークと，それと同規模の単一 OCS ネットワークからなるネットワーク環境，およびその環境での通信手法を提案した[10)]．本研究では，OCS ネットワークの規模を縮小し，さらに OCS ノードの EPS リンクバンド幅を増やすことで中継負荷削減を行った．通信手法については，一部のノードのみ OCS ネットワークに接続されている制約を考慮して光回線割当てを行った点，EPS スイッチ間通信数を考慮して通信数の多いスイッチ間に光回線を多く割り当てた点が異なる．小規模 OCS ネットワークを補助的に使用することで，過去の研究，および，フルバイセクションバンド幅 EPS ネットワークと同程度の性能を示すことを実証した．

アプリケーションの通信パターンを活かした通信最適化手法には，プロセスマイグレーションを行うもの[14)]，プロセスをネットワーク上に最適配置するもの[15),16)] がある．しか

しながら，プロセスマイグレーション手法では移動先プロセスの決定にかかるコスト以外にも，メモリイメージ転送コストがかかる．今後システムがより大規模化されるにつれ，移動するプロセス数や，使用メモリ量も増加していくと考えられるので，転送コストは大きくなり，効果が薄れる．Dixit-Radiya ら[15] や，Bhanot ら[16] はトポロジの固定されたネットワーク上で，アプリケーション通信パターンとネットワーク通信コストからプロセスの最適配置を求めている．しかしながら，プロセス最適配置手法だけではネットワークトポロジに制約されるため，通信パターンを活かしきれない．トポロジの再構成が可能なネットワークと組み合わせることで，より効果を発揮する手法である．

7. ま と め

大規模 HPC システム用のネットワークとして，EPS ネットワークを使用するシステムで OCS ネットワークを補助的に使用すること，および，その環境でのアプリケーション通信パターンを考慮した MPI 通信手法を提案した．OCS ネットワークの補助的利用とは，EPS ネットワークに接続されたノードの一部を単一小規模 OCS ネットワークへも接続することである．提案通信手法では，アプリケーションの通信パターンを可能な限り満たすように，OCS ネットワークに接続されたノード間に光回線を割り当て，それらノードが EPS スイッチをまたぐ大容量メッセージを中継転送する．評価の結果，EPS ネットワークのバイセクションバンド幅が低い場合でも，全ノードの半数だけを OCS ネットワークに接続することで，高いバイセクションバンド幅を要求するアプリケーションの実行性能の向上を確認した．特に，全対全通信を行うアプリケーションに対して，Fat Tree EPS ネットワークと同程度の性能を示すことを確認した．

今後の課題として，光回線割当てアルゴリズムの改良を考えている．今回の提案手法では，光回線を割り当てる際に条件を満たす複数の OCS ノード間通信がある場合，単純に通信量により選択している．この場合，各 OCS ノードは 1 つの光回線しか使用できないため，他の重要な通信に光回線を割り当てることができず，性能低下となる可能性がある．アプリケーション全体の通信状況を考慮して光回線を割り当てるように変更したいと考えている．さらに，より多くのアプリケーションでの評価，数千プロセス規模の大規模環境での評価を行う．

謝辞　本研究の一部は科学研究費補助金特定領域研究（18049028），および，JSPS グローバル COE プログラム「計算世界観の深化と展開」の補助による．

参 考 文 献

1) Davis, K., Hoisie, A., Johnson, G., Kerbyson, D.J., Lang, M., Pakin, S. and Petrini, F.: A Performance and Scalability Analysis of the BlueGene/L Architecture, *SC '04: Proc. 2004 ACM/IEEE conference on Supercomputing*, Washington, DC, USA, IEEE Computer Society (2004).
2) Matsuoka, S.: The Road to TSUBAME and Beyond, *High Performance Computing on Vector Systems 2007*, Vol.6, pp.265–267 (2007).
3) TACC HPC Systems. http://www.tacc.utexas.edu/resources/hpcsystems/
4) T2K Open Supercomputer Alliance. http://www.open-supercomputer.org/
5) Shao, S., Jones, A.K. and Melhem, R.: A Compiler-based Communication Analysis Approach for Multiprocessor Systems, *Proc. 20th IEEE International Parallel and Distributed Processing Symposium* (2006).
6) Barker, K.J., Benner, A., Hoare, R., Hoisie, A., Jones, A.K., Kerbyson, D.J., Li, D., Melhem, R., Rajamony, R., Schenfeld, E., Shao, S., Stunkel, C. and Walker, P.: On the Feasibility of Optical Circuit Switching for High Performance Computing Systems, *Proc. 2005 ACM/IEEE conference on Supercomputing* (2005).
7) Barker, K.J. and Kerbyson, D.J.: Performance Analysis of an Optical Circuit Switched Network for Peta-Scale Systems, *Euro-Par 2007*, pp.858–867 (2007).
8) Kamil, S., Pinar, A., Gunter, D., Lijewski, M., Oliker, L. and Shalf, J.: Reconfigurable Hybrid Interconnection for Static and Dynamic Scientific Applications, *ACM International Conference on Computing Frontiers* (2007).
9) Dobbelaere, P.D., Falta, K., Fan, L., Gloeckner, S. and Patra, S.: Degital MEMS for Optical Switching, *Communications Magazine*, Vol.40, pp.88–95, IEEE (2002).
10) 滝澤真一朗，遠藤敏夫，松岡　聡：次世代光インターコネクトでの MPI 通信に関する研究，コンピュータソフトウェア (2008). (to appear)
11) der Wijngaart, R.F.V.: NAS Parallel Benchmarks Version 2.4, Technical Report NAS Technical Report NAS-02-007, NASA Ames Research Center (2002).
12) METIS—Family of Multilevel Partitioning Algorithms. http://glaros.dtc.umn.edu/gkhome/views/metis/
13) Thakur, R., Rabenseifner, R. and Gropp, W.: Optimization of Collective Communication Operations in MPICH, *International Journal of High Performance Computer Applications*, Vol.19, No.1, pp.49–66 (2005).
14) Maghraoui, K.E., Desell, T., Szymanski, B.K., Teresco, J.D. and Varela, C.: Towards a Middleware Framework for Dynamically Reconfigurable Scientific Computing, Grid Computing and New Frontiers of High Performance Processing, Grandinetti, L. (Ed.), *Advances in Parallel Computing*, Vol.14, pp.275–301, Elsevier (2005).

15) Dixit-Radiya, V.A. and Panda, D.K.: Task Assignment on Distributed-Memory Systems with Adaptive Wormhole Routing, *The 5th IEEE Symposium on Parallel and Distributed Processing*, pp.674–681 (1993).
16) Bhanot, G., Gara, A., Heidelberger, P., Lawless, E., Sexton, J.C. and Walkup, R.: Optimizing task layout on the Blue Gene/L supercomputer, *IBM Journal of Research and Development*, Vol.49, No.2/3, pp.489–500 (2005).

(平成 20 年 10 月 3 日受付)
(平成 21 年 2 月 2 日採録)

滝澤真一朗（正会員）

1981 年生．2009 年東京工業大学大学院数理・計算科学専攻博士課程修了．博士（理学）．東京工業大学グローバル COE「計算世界観の深化と展開」RA を経て，2009 年より東京工業大学学術国際情報センター産学官連携研究員．主に分散並列環境での通信最適化の研究に従事．日本ソフトウェア科学会会員．

遠藤　敏夫（正会員）

1974 年生．2001 年東京大学大学院理学系研究科情報科学専攻博士課程修了．博士（理学）．東京大学情報理工学系研究科特任助手，東京工業大学学術国際情報センター特任講師等を経て，2007 年より東京工業大学グローバル COE「計算世界観の深化と展開」特任准教授．主に分散処理やヘテロ型アーキテクチャ上での並列計算の研究に従事．日本ソフトウェア科学会，ACM，IEEE-CS 各会員．

松岡　　聡（正会員）

1986 年東京大学理学部情報科学科卒業，1989 年同大学大学院博士課程から，学情報科学科助手に採用，同大学情報工学専攻講師を経て，1996 年に東京工業大学情報理工学研究科数理・計算科学専攻助教授．2001 年 4 月に東京工業大学学術国際情報センター教授，2002 年より国立情報学研究所の客員教授を併任．博士（理学）（東京大学）．高性能システム，並列処理，グリッド計算，クラスタ計算機，高性能・並列オブジェクト指向言語処理系，等の研究に従事．わが国のナショナルグリッドプロジェクトである NAREGI プロジェクトのサブリーダーであり，また 2006 年時点でわが国最高性能のスーパコンピュータ TSUBAME を構築．1996 年度情報処理学会論文賞，1999 年情報処理学会坂井記念賞，2006 年学術振興会賞（JSPS Award）等を受賞．国際学会 ISOTAS'96，ECOOP'97，ISCOPE'99，ACM OOPSLA'02，IEEE CCGrid'03，HPC Asia 05，Grid2006 等のプログラム（副）委員長，IEEE/ACM Supercomputing'04–Network Track Chair，Reflection'01，IEEE CCGrid'06 大会委員長．また，グリッド国際標準化団体の Global Grid Forum の Steering Group 委員等を務める．